IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of	)	
	)	**For: TURBO DECODING METHOD AND APPARATUS FOR WIRELESS COMMUNICATIONS**
Sindhushayana et al.	)	
	)	
	)	
Serial No. 09/965,518	)	
	)	
Filed: September 25, 2001	)	**Group No. 2611**

TRANSMITTAL LETTER

Mail Stop Appeal Briefs - Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

In response to the Notice of Appeal filed concurrently herewith, Applicants request a one-month extension of time from June 29, 2006 to July 29, 2006, enclosed is:

CERTIFICATE OF MAILING/TRANSMISSION (37 CFR 1.8(a))

I hereby certify that this correspondence is, on the date shown below, being:

MAILING

☒ deposited with the United States Postal Service with sufficient postage as first class mail, in an envelope addressed to the Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.

Depositor's Name: Peggy Bozym
(type or print name)

Date: July 31, 2006

FACSIMILE

☐ transmitted by facsimile to the Patent and Trademark Office.

Depositor's Name: _____
(type or print name)

Signature: Peggy Bozym

(Customer No. 23696)

Please charge Deposit Account No. 17-0026 of QUALCOMM Incorporated in the amount of $500.00 to pay the necessary fee due under 37 CFR § 1.17(b) for the filing of the Notice of Appeal. The Commissioner is hereby authorized to charge any additional fees that may be required, or credit any overpayment, to said Deposit Account No. 17-0026. A duplicate copy of this notice is enclosed.

Respectfully submitted,

Dated: July 31, 2006

By: _____
Larry J. Moskowitz, Reg. No. 42,911
(858) 651-4536

QUALCOMM Incorporated
Attn: Patent Department
5775 Morehouse Drive
San Diego, California 92121-1714
Telephone: (858) 658-5787
Facsimile: (858) 658-2502

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In Re Application No. 09/965,518	**For:**	**Turbo Decoding Method and Apparatus for Wireless Communications**
Sindhushayana et al.		
Examiner: Juan A. Torres		
Filed: September 25, 2001	**Group No.**	**2611**

APPEAL BRIEF TO THE BOARD OF PATENT APPEALS AND INTERFERENCES

Mail Stop Appeal Brief-Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

This Appeal Brief is responsive to the rejections in Final Office Action mailed on March 29, 2006. It is being filed within two months of receipt by the Office of the Notice of Appeal, and therefore is timely. If the undersigned attorney is mistaken in this regard, Applicants petition for an extension of time and authorization is hereby granted to charge all required time extension fees to Deposit Account No. 17-0026. Authorization is also granted to charge to the same Deposit Account the Appeal Brief fee (37 C.F.R. § 41.20(b)(2)) and all other fees necessary to file this Appeal Brief.

CERTIFICATE OF MAILING/TRANSMISSION (37 CFR 1.8(a))

I hereby certify that this correspondence is, on the date shown below, being:

MAILING

☒ deposited with the United States Postal Service with sufficient postage as first class mail, in an envelope addressed to the Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.

Depositor's Name: Peggy Bozym
(type or print name)

Date: July 31, 2006

Signature: Peggy Bozym

FACSIMILE

☐ transmitted by facsimile to the Patent and Trademark Office.

Depositor's Name:
(type or print name)

Date:

Signature: _____

Attorney Docket No.: 000400
Customer No.: 23696

I
REAL PARTY IN INTEREST

In this Appeal, the real party in interest is Qualcomm Incorporated, a Delaware corporation, having a place of business at 5775 Morehouse Drive, San Diego, California 92121.

II
RELATED APPEALS AND INTERFERENCES

Appellants, Assignee, and the undersigned attorney do not know of any other appeal, interference, or judicial proceeding that is related to, directly affects, is directly affected by, or has a bearing on the decision of the Board of Patent Appeals and Interferences in this Appeal.

III
STATUS OF CLAIMS

The status of claims in the instant application is as follows:

Claims 1-3, 6, 7, 9-18, and 20-24 have been rejected and are pending.

Applicants appeal from the rejection of claims 1-3, 6, 7, 9-18, and 20-24.

IV
STATUS OF AMENDMENTS

An amendment was filed on May 30, 2006, after the rejection of claims in the Final Office Action mailed on March 29, 2006. According to the Advisory Action mailed on June 23, 2006, the amendment was not entered.

V
SUMMARY OF CLAIMED SUBJECT MATTER

A. Independent Claims

Claim 1

Claim 1 is directed to a method used in a communication system for decoding a sequence of turbo encoded data symbols transmitted over a channel. *E.g.*, Specification at page 2, lines 2-19 (par. 0004); *id.* at page 4, lines 9-21 (par. 00014); *id.* at page 4, line 22 through page 5, line 11 (par. 00015).[1]

The method includes updating channel nodes R_x, R_y and R_z (for example, channel nodes of a turbo decoder) based on a received channel output. *E.g.*, Specification at page 2, lines 2-19 (par. 0004); *id.* at page 15, lines 1-18 (par. 00034); *id.* at page 20, lines 17-22 (par. 00046).

The method further includes initializing outgoing messages from symbol nodes X_i, Y_i and Z_k, wherein the symbol nodes X_i, Y_i and Z_k are in communication with the channel nodes R_x, R_y and R_z. *E.g.*, Specification at page 2, lines 2-19 (par. 0004); Fig. 7, elements R_x, 701, R_y, 707, R_z, 708; the Abstract.

The method further includes triggering updates of computational nodes C (for example, of a first constituent decoder included in the turbo decoder) and D (for example, of a second constituent decoder included in the turbo decoder), associated with different instances of time, in accordance with a triggering schedule. *E.g.*, Specification at page 2, lines 2-19 (par. 0004); *id.* at page 21, line 11 through page 22, line 12 (par. 00048).

A computational node C_i (of the nodes C) is in communication with the symbol nodes X_i and Y_i, and a computational node D_k (of the nodes D) is in communication with the symbol

[1] Please note that the paragraphs cited are those of the Application as filed, not of the published Application. Paragraph numbers in the two sources diverge beginning with paragraph 00034.

nodes X_i and Z_k. *E.g.*, Fig. 7, elements 701, 704, 706, 707, 708.

The triggering schedule includes triggering all the computational nodes C and D at different instances of time essentially concurrently for each decoding iteration. *E.g.*, Specification at page 21, line 11 through page 22, line 12 (par. 00048).

Claim 6

Claim 6 is directed to a method used in a communication system for decoding a sequence of turbo encoded data symbols transmitted over a channel. *E.g.*, Specification at page 2, lines 2-19 (par. 0004); *id.* at page 4, lines 9-21 (par. 00014); *id.* at page 4, line 22 through page 5, line 11 (par. 00015).

The method includes updating channel nodes R_x, R_y and R_z (for example, channel nodes of a turbo decoder) based on a received channel output. *E.g.*, Specification at page 2, lines 2-19 (par. 0004); *id.* at page 15, lines 1-18 (par. 00034); *id.* at page 20, lines 17-22 (par. 00046).

The method further includes initializing outgoing messages from symbol nodes X_i, Y_i and Z_k, wherein the symbol nodes X_i, Y_i and Z_k are in communication with the channel nodes R_x, R_y and R_z. *E.g.*, Specification at page 2, lines 2-19 (par. 0004); Fig. 7, elements R_x, 701, R_y, 707, R_z, 708; the Abstract.

The method further includes triggering updates of computational nodes C (for example, of a first constituent decoder included in the turbo decoder) and D (for example, of a second constituent decoder included in the turbo decoder), associated with different instances of time, in accordance with a triggering schedule. *E.g.*, Specification at page 2, lines 2-19 (par. 0004); *id.* at page 21, line 11 through page 22, line 12 (par. 00048).

A computational node C_i (of the nodes C) is in communication with the symbol nodes X_i and Y_i, and a computational node D_k (of the nodes D) is in communication with the symbol

nodes X_i and Z_k. *E.g.*, Fig. 7, elements 701, 704, 706, 707, 708.

The method further includes partitioning the computational node C at time instances C_0, C_1, C_2, ..., C_N into at least two subsets, wherein the triggering schedule includes triggering updates of computational nodes C in a sequence at different time instances in each subset, and wherein the triggering of computational node C at different time instances in the least two subsets occurs concurrently. For example, the computational nodes C_0, C_1, C_2, ..., C_N may be divided into a first subset (first sub-block) and a second subset (second sub-block). The nodes C in the first subset are triggered sequentially, as are the nodes C in the second subset; but the triggering is performed concurrently in the two subsets. Thus, two nodes C (one from each of the two subsets) may be triggered at the same time. *E.g.*, Specification at page 22, lines 13-22 (par. 00049).

Claim 18

Claim 18 is directed to an apparatus for decoding a sequence of turbo encoded data symbols communicated over a channel. *E.g.*, Specification at page 2, lines 2-19 (par. 0004); *id.* at page 4, lines 9-21 (par. 00014); *id.* at page 4, line 22 through page 5, line 11 (par. 00015).

The apparatus includes channel nodes R_x, R_y and R_z (for example, channel nodes of a turbo decoder) for receiving channel output. *E.g.*, Specification at page 2, lines 2-19 (par. 0004); *id.* at page 15, lines 1-18 (par. 00034); *id.* at page 20, lines 17-22 (par. 00046).

The apparatus further includes symbol nodes X_i, Y_i and Z_k in communication with the channel nodes R_x, R_y and R_z. Specification at page 2, lines 2-19 (par. 0004); *id.* at page 21, line 11 through page 22, line 12 (par. 00048); Fig. 7, elements R_x, 701, R_y, 707, R_z, 708.

The apparatus further includes state nodes S_i and S_{i-1} associated with a first constituent code in a turbo code. *E.g.*, Specification at page 2, lines 2-19 (par. 0004); *id.* at page 21, line 11

through page 22, line 12 (par. 00048); Fig. 7, elements 702.

The apparatus further includes state nodes σ_k and σ_{k-1} associated with a second constituent code in the turbo code. *E.g.*, Specification at page 2, lines 2-19 (par. 0004); *id.* at page 21, line 11 through page 22, line 12 (par. 00048); Fig. 7, elements 705.

The apparatus further includes a computational node C_i (for example, of a first constituent decoder within the turbo decoder) in communication with the symbol nodes X_i and Y_i, and a computational node D_k (for example, of a second constituent decoder within the turbo decoder) in communication with the symbol nodes X_i and Z_k, wherein the computational node C_i is in communication with the state nodes S_i and S_{i-1} and the computational node D_k is in communication with the state nodes σ_k and σ_{k-1}. *E.g.*, Specification at page 21, line 11 through page 22, line 12 (par. 00048); Fig. 7, elements 702, 704, 705, 706.

The apparatus further includes a computational node C_{i+1} that is in communication with the state node S_i, and a computational node C_{i-1} that is in communication with the state node S_{i-1}. *E.g.*, Specification at page 21, line 11 through page 22, line 12 (par. 00048); Fig. 7, elements 704.

The apparatus further includes a computational node D_{K+1} that is in communication with the state node σ_k, and a computational node D_{k-1} that is in communication with the state node σ_{k-1}. *E.g.*, Specification at page 21, line 11 through page 22, line 12 (par. 00048); Fig. 7, elements 706.

The computational nodes C and D at different time instances are configured for updates in accordance with an update triggering schedule, which schedule includes concurrent triggering of each node of a first plurality of the computational nodes C, and concurrent triggering of each node of a second plurality of computational nodes D. *E.g.*, Specification at page 22, lines 13

through 22 (par. 00049).

Claim 22

Claim 22 is directed to a processor configured for decoding a sequence of turbo encoded data symbols for communication over a channel. *E.g.*, Specification at page 2, lines 2-19 (par. 0004); *id.* at page 4, lines 9-21 (par. 00014); *id.* at page 4, line 22 through page 5, line 11 (par. 00015).

The processor is configured to include channel nodes R_x, R_y and R_z (for example, channel nodes of a turbo decoder) for receiving channel output. *E.g.*, Specification at page 2, lines 2-19 (par. 0004); *id.* at page 15, lines 1-18 (par. 00034); *id.* at page 20, lines 17-22 (par. 00046).

The processor is further configured to include symbol nodes X_i, Y_i and Z_k in communication with the channel nodes R_x, Ry and R_z. Specification at page 2, lines 2-19 (par. 0004); *id.* at page 21, line 11 through page 22, line 12 (par. 00048); Fig. 7, elements R_x, 701, R_y, 707, R_z, 708.

The processor is further configured to include state nodes S_i and S_{i-1} associated with a first constituent code in a turbo code. *E.g.*, Specification at page 2, lines 2-19 (par. 0004); *id.* at page 21, line 11 through page 22, line 12 (par. 00048); Fig. 7, elements 702.

The processor is further configured to include state nodes σ_k and σ_{k-1} associated with a second constituent code in the turbo code. *E.g.*, Specification at page 2, lines 2-19 (par. 0004); *id.* at page 21, line 11 through page 22, line 12 (par. 00048); Fig. 7, elements 705.

The processor is further configured to include a computational node C_i (for example, of a first constituent decoder within the turbo decoder) in communication with the symbol nodes X_i and Y_i, and a computational node D_k (for example, of a second constituent decoder within the turbo decoder) in communication with the symbol nodes X_i and Z_k, wherein the computational

node C_i is in communication with the state nodes S_i and S_{i-1}, and the computational node D_k is in communication with the state nodes σ_k and σ_{k-1}. *E.g.*, Specification at page 21, line 11 through page 22, line 12 (par. 00048); Fig. 7, elements 702, 704, 705, 706.

The processor is further configured to include a computational node C_{i+1} in communication with the state node S_i, and a computational node C_{i-1} in communication with the state node S_{i-1}. *E.g.*, Specification at page 21, line 11 through page 22, line 12 (par. 00048); Fig. 7, elements 704.

The processor is further configured to include a computational node D_{K+1} in communication with the state node σ_k, and a computational node D_{k-1} in communication with the state node σ_{k-1}. *E.g.*, Specification at page 21, line 11 through page 22, line 12 (par. 00048); Fig. 7, elements 706.

The computational nodes C and D at different time instances are configured for updates in accordance with an update triggering schedule, which schedule includes concurrent triggering of each node of a first plurality of the computational nodes C, and concurrent triggering of each node of a second plurality of computational nodes D. *E.g.*, Specification at page 22, lines 13 through 22 (par. 00049).

VI
CONCISE STATEMENT OF THE GROUNDS OF REJECTION

1. Claims 1-3, 6, 7, 9-14, 17, 18, and 20-24 stand rejected under 35 U.S.C. § 102 as being anticipated by admitted prior art in Figure 5 of the present Application.

2. Claims 15 and 16 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over the admitted prior art in Figure 5 of the present Application in view of Xu *et al.*, U.S. Patent Application Publication Number 2001/0052104 ("Xu").

VII
ARGUMENT

A. Rejection of Independent Claim 1 Under Section 102

For convenience of discussion, claim 1 is set forth below:

> 1. In a communication system, a method for decoding a sequence of turbo encoded data symbols transmitted over a channel comprising:
>
> updating channel nodes R_x, R_y and R_z based on a received channel output;
>
> initializing outgoing messages from symbol nodes X_i, Y_i and Z_k, wherein said symbol nodes X_i, Y_i and Z_k are in communication with said channel nodes R_x, R_y and R_z; and
>
> triggering updates of computational nodes C and D, associated with different instances of time, in accordance with a triggering schedule, wherein a computational node C_i is in communication with said symbol nodes X_i and Y_i and a computational node D_k is in communication with said symbol nodes X_i and Z_k;
>
> wherein said triggering schedule includes triggering all said computational nodes C and D at different instances of time essentially concurrently for each decoding iteration.

The Final Office Action asserted (at pages 8-9) that the admitted prior art in Figure 5 anticipates this claim.

The legal standard for claim anticipation is well established. "A claim is anticipated only if each and every element as set forth in the claim is found, either expressly or inherently described, in a single prior art reference." *Verdegaal Bros. v. Union Oil Co. of California*, 814 F.2d 628, 631, 2 U.S.P.Q.2d 1051, 1053 (Fed. Cir. 1987). "The identical invention must be shown in as complete detail as is contained in the . . . claim." *Richardson v. Suzuki Motor Co.*, 868 F.2d 1226, 1236, 9 U.S.P.Q.2d 1913, 1920 (Fed. Cir. 1989). (Both *Verdegaal* and *Richardson* opinions are quoted with approval in MPEP § 2131.) We respectfully submit that the admitted prior art in Figure 5 does not describe, either expressly or inherently, the limitation of triggering all said computational nodes C and D at different instances of time essentially

concurrently for each decoding iteration.

Figure 5 is a block diagram of a decoder 500, such as a turbo decoder. The structure and operation of the decoder 500 are described in the instant Application as follows:

> Referring to Fig. 5, a block diagram of a decoder 500 is shown for decoding the noisy version of data symbols X_i, Y_i and Z_k to produce estimates of data symbols X_i. Decoder 500 may be used in decoder 20 of system 10. The noisy version of encoded data symbols X_i, Y_i and Z_k may pass through a data symbol selector block 520 which operates to select the noisy version of data symbols X_i and Y_i for routing to a decoder block 501 at an input 542. The noisy version of data symbols X_i internally passes through an interleaver 599 to locally reproduce a noisy version of data symbols X_k. The locally produced noisy version of data symbols X_k and Z_k pass to a decoder block 502 at an input 540. Decoder 501 may decode the noisy version of data symbols X_i and Y_i according to a decoding process such as MAP as explained and shown. Decoder 501 produces estimates of data symbols X_i at an output 550. Decoder 502 decodes the noisy version of data symbols Z_k and X_k according to a decoding process such as MAP as explained and shown. Decoder 502 produces estimates of data symbols X_k at an output 560. The decoding processes in decoders 501 and 502 may be performed sequentially. The information may pass from decoder 501 to decoder 502 after completing each iteration. One ordinary skilled in the art may appreciate that the decoders in various embodiments as described and shown operate on the noisy version of the encoded data symbols.
>
> To increase confidence for the estimate of the data symbols X_i to hold a true value, the estimate of data symbols X_i at output 550 may pass through an interleaver 530 to produce estimates of data symbols X_k at an input 532 of decoder 502. Decoder 502 uses the estimates of data symbols X_k at input 532 with estimates of data symbols at input 540 to produce new estimates of data symbols X_k at output 560. Estimates of data symbols X_k at output 560 pass through a de-interleaver 531 to reverse the process of interleaving function of interleaver 430 in the turbo code 400, and to produce estimates of data symbols X_i at an input 541. Estimates of data symbols X_i at input 541 are used with the estimates of data symbols at input 542 to produce a new estimate of data symbols X_i at output 550. The process may be repeated until confidence for the estimate of data symbols X_i reaches an acceptable level. As such, the process for decoding data symbols X_i is time consuming and may not produce an accurate estimate of the data symbols X_i in a timely manner for an application in the communication system 10.

Specification, at page 10, line 6 through page 11, line 22, pars. 00027 and 00028.

The decoder 500 thus includes a selector block 520, constituent decoders 501 (the first decoder) and 502 (the second decoder), an interleaver 530, and a de-interleaver 531. Figure 5

shows one detail of the selector block 520, which is an interleaver 599. Notably, no details are shown for the first and second constituent decoders 501 and 502 – each decoder 501/502 is drawn as a block. Even if it is understood that the constituent decoders 501 and 502 include computational nodes or similar elements, and that such computational nodes are "triggered," there is nothing in either Figure 5 or the description of the decoder 500 that would specify the particular order in which the computational nodes are triggered within each of the constituent decoders 501 and 502 for each iteration. As regards the order of operation of the constituent decoders 501 and 502, the specification states that "[t]he decoding processes in decoders 501 and 502 may be performed sequentially." Specification at page 11, lines 3-4 (par. 00028) (emphasis added).

Neither Figure 5 nor the description of the operation of the decoder 500 shown in Figure 5 specifies the order of triggering of the computational nodes in each iteration, with the exception that the decoding processes in decoders 501 and 502 may be performed sequentially. In contrast, claim 1 requires *triggering all said computational nodes C and D at different instances of time essentially concurrently for each decoding iteration.*

The meaning of "essentially concurrently" is explained in the specification of the present application. In particular, the specification explains the meaning of "essentially concurrently" thus:

> In accordance with an embodiment, all the computation nodes 704 and 706 may be triggered essentially concurrently. As such, in one step all the computational nodes are once updated. Each time all the computational nodes are updated, the decoding process may have completed on [sic, one] decoding iteration. The concurrent trigger of the computational nodes 704 and 706 may be repeated to achieve one or more iterations of the decoding process.

Specification, at page 22, lines 6-12 (last four sentences of par. 00048) (emphasis added). Thus, "essentially concurrent" triggering of all computational nodes means that the nodes are updated

in one step to complete one decoding iteration; the results of updating of some nodes are not used in updating other nodes during the same iteration in order to perform the updating in one step. Figure 5 and its description in the specification do not disclose this limitation. Furthermore, the Final Office Action did not assert that such triggering is inherent. Indeed, such concurrent triggering is not inherent, for the nodes may be triggered sequentially.

Applicants respectfully submit that the admitted prior art in Figure 5 does not anticipate claim 1 at least because the admitted prior art does not disclose that the computational nodes are triggered essentially concurrently for each decoding iteration.

B. Rejection of Independent Claim 6 Under Section 102

For convenience of discussion, independent claim 6 is set forth below:

> 6. In a communication system, a method for decoding a sequence of turbo encoded data symbols transmitted over a channel comprising:
>
> updating channel nodes R_x, R_y and R_z based on a received channel output;
>
> initializing outgoing messages from symbol nodes X_i, Y_i and Z_k, wherein said symbol nodes X_i, Y_i and Z_k are in communication with said channel nodes R_x, R_y and R_z;
>
> triggering updates of computational nodes C and D, associated with different instances of time, in accordance with a triggering schedule, wherein a computational node C_i is in communication with said symbol nodes X_i and Y_i and a computational node D_k is in communication with said symbol nodes X_i and Z_k; and
>
> partitioning said computational node C at time instances C_0, C_1, C_2, ..., C_N into at least two subsets, wherein said triggering schedule includes triggering updates of computational nodes C in a sequence at different time instances in each subset, and wherein said triggering of computational node C at different time instances in said least two subsets occurs concurrently.

In accordance with the method of this claim, the nodes C_0, C_1, C_2, ..., C_N are partitioned into at least two subsets or sub-blocks, and the nodes in the different subsets are triggered

concurrently, while the nodes in the same subset are triggered sequentially. Consider an example in which a first subset includes nodes C_0 and C_1, and the second subset includes nodes C_2 and C_3. In this example, the nodes C_0 and C_1 of the first subset are triggered in sequence, first C_0 and then C_1; similarly, the nodes C_2 and C_3 are also triggered in sequence, first C_2 and then C_3. But the nodes of the two subsets are triggered concurrently in this example; thus, the nodes C_0 and C_2 are triggered concurrently or at the same time (for example, at a time T_a), and the nodes C_1 and C_3 are also triggered concurrently or at the same time (for example, at a time T_b). Therefore, multiple computational nodes are triggered at the same time.

As discussed in more detail above in relation to claim 1, Figure 5 and its description do not teach concurrent triggering of multiple computational nodes. Furthermore, Figure 5 and its description fail to disclose the step of partitioning. At least for these reasons, Applicants respectfully submit that the admitted prior art of Figure 5 does not anticipate independent claim 6.

C. Rejection of Independent Claims 18 and 22 Under Section 102

Each of these independent claims recites, *inter alia*, the limitations of "wherein computational nodes C and D at different time instances are configured for updates in accordance with an update triggering schedule, *said update triggering schedule including concurrent triggering of each node of a first plurality of said computational nodes C*, and *concurrent triggering of each node of a second plurality of computational nodes D*." (Italicization has been added for emphasis in the immediately preceding quotation from claims 18 and 22.) Each of these claims therefore also requires concurrent triggering of a plurality of computational nodes. As discussed in more detail above in relation to claim 1, the admitted prior art of Figure 5 does not disclose concurrent triggering of a plurality of computational nodes. At least for this reason,

Applicants respectfully submit that the admitted prior art of Figure 5 does not anticipate independent claims 18 and 22.

D. Rejection of Dependent Claims

The above discussion addresses rejections of all independent claims of the application. Dependent claims should be patentable at least for the same reasons as their base claims and intervening claims, if any.

VIII
CONCLUSION

In view of the foregoing, Applicants submit that all pending claims in the application are patentable and request reversal of the rejections.

Respectfully submitted,

Dated: **July 31, 2006**

By: _____
Larry Moskowitz, Reg. No. 42,911
(858) 651-4556

QUALCOMM Incorporated
5775 Morehouse Drive
San Diego, California 92121
Telephone: (858) 658-5787
Facsimile: (858) 658-2502

CLAIMS APPENDIX

The following is a listing of the claims in the application. All claims have been rejected and are involved in this Appeal.

1. (Previously Presented) In a communication system, a method for decoding a sequence of turbo encoded data symbols transmitted over a channel comprising:

updating channel nodes R_x, R_y and R_z based on a received channel output;

initializing outgoing messages from symbol nodes X_i, Y_i and Z_k, wherein said symbol nodes X_i, Y_i and Z_k are in communication with said channel nodes R_x, R_y and R_z; and

triggering updates of computational nodes C and D, associated with different instances of time, in accordance with a triggering schedule, wherein a computational node C_i is in communication with said symbol nodes X_i and Y_i and a computational node D_k is in communication with said symbol nodes X_i and Z_k;

wherein said triggering schedule includes triggering all said computational nodes C and D at different instances of time essentially concurrently for each decoding iteration.

2. (Original) The method as recited in claim 1 wherein said computational node C_i is in communication with state nodes S_i and S_{i-1} associated with a first constituent code, and said computational node D_k is in communication with state nodes σ_k and σ_{k-1} associated with a second constituent code, wherein said first and second constituent codes are associated with a turbo code in said communication system used for encoding said sequence of encoded data symbols.

3. (Original) The method as recited in claim 1 further comprising:

accepting a value of symbol X_i at said symbol node X_i as a decoded value of symbol X_i after at least one iteration of said triggering updates of said computational nodes C and D.

4. (Canceled)

5. (Canceled)

6. (Previously Presented) In a communication system, a method for decoding a sequence of turbo encoded data symbols transmitted over a channel comprising:

updating channel nodes R_x, R_y and R_z based on a received channel output;

initializing outgoing messages from symbol nodes X_i, Y_i and Z_k, wherein said symbol nodes X_i, Y_i and Z_k are in communication with said channel nodes R_x, R_y and R_z;

triggering updates of computational nodes C and D, associated with different instances of time, in accordance with a triggering schedule, wherein a computational node C_i is in communication with said symbol nodes X_i and Y_i and a computational node D_k is in communication with said symbol nodes X_i and Z_k; and

partitioning said computational node C at time instances C_0, C_1, C_2, ..., C_N into at least two subsets, wherein said triggering schedule includes triggering updates of computational nodes C in a sequence at different time instances in each subset, and wherein said triggering of computational node C at different time instances in said at least two subsets occurs concurrently.

7. (Original) The method as recited in claim 6 further comprising:

determining said sequence at different time instances in each subset for said triggering updates.

8. (Canceled)

9. (Original) The method as recited in claim 6 wherein said least two subsets of computational node C at different time instances C_0, C_1, C_2, ..., C_N have at least one common computational node time instance.

10. (Previously Presented) The method as recited in claim 6 further comprising:

partitioning computational node D at different time instances D_0, D_1, D_2, ..., D_N into at least two subsets, wherein said triggering schedule includes triggering computational nodes D at different time instances in a sequence in each subset.

11. (Original) The method as recited in claim 10 further comprising:

determining said sequence at different time instances in each subset for said triggering updates.

12. (Original) The method as recited in claim 10 wherein said triggering of computational node D at different time instance in said least two subsets occurs concurrently.

13. (Original) The method as recited in claim 10 wherein said subsets of computational node D at time instances D_0, D_1, D_2, ..., D_N have at least one common computational node time instance.

14. (Original) The method as recited in claim 1 wherein said updating includes summing incoming messages to produce an output message, and outputting said output message for updating.

15. (Previously Presented) The method as recited in claim 1 wherein said updating said channel nodes R_x, R_y and R_z based on said received channel output includes:

receiving at said channel node R_x said channel output associated with a symbol X_i;

receiving at said channel node R_y said channel output associated with a symbol Y_i;

receiving at said channel node R_z said channel output associated with a symbol Z_k;

passing from said channel node R_x a likelihood of said symbol X_i, based on said received channel output, to said symbol node X_i;

passing from said channel node R_y a likelihood of said symbol Y_i, based on said received channel output, to said symbol node Y_i; and

passing from said channel node R_z a likelihood of said symbol Z_k, based on said received channel output, to said symbol node Z_k.

16. (Original) The method as recited in claim 1 wherein said initializing outgoing messages from symbol nodes X_i, Y_i and Z_k includes:

passing a message from said symbol node X_i to said computational node C_i of said computational node C, wherein said message is based on a summation of incoming messages at said symbol node X_i;

passing a message from said symbol node X_i to said computational node D_k of said computational node D, wherein said message is based on a summation of incoming messages at said symbol node X_i;

passing a message from said symbol node Y_i to said computational node C_i, wherein said message is based on said likelihood of data symbol Y_i; and

passing a message from said symbol node Z_k to said computational node D_k, wherein said message is based on said likelihood of data symbol Z_k.

17. (Original) The method as recited in claim 1 wherein said sequence of data includes "N" number of symbols, wherein each symbol in said sequence is identified by either a subscript "i" or "k," and wherein said subscript "i" and "k" are references to time instances in the decoding process.

18. (Previously Presented) An apparatus for decoding a sequence of turbo encoded data symbols communicated over a channel comprising:

channel nodes R_x, R_y and R_z for receiving channel output;

symbol nodes X_i, Y_i and Z_k in communication with said channel nodes R_x, R_y and R_z;

state nodes S_i and S_{i-1} associated with a first constituent code in a turbo code;

state nodes σ_k and σ_{k-1} associated with a second constituent code in said turbo code;

a computational node C_i in communication with said symbol nodes X_i and Y_i;

a computational node D_k in communication with said symbol nodes X_i and Z_k, wherein said computational node C_i is in communication with said state nodes S_i and S_{i-1} and said computational node D_k is in communication with said state nodes σ_k and σ_{k-1};

a computational node C_{i+1} in communication with said state node S_i;

a computational node C_{i-1} in communication with said state node S_{i-1};

a computational node D_{K+1} in communication with said state node σ_k; and

a computational node D_{k-1} in communication with said state node σ_{k-1};

wherein computational nodes C and D at different time instances are configured for updates in accordance with an update triggering schedule, said update triggering schedule including concurrent triggering of each node of a first plurality of said computational nodes C, and concurrent triggering of each node of a second plurality of computational nodes D.

19. (Canceled)

20. (Original) The apparatus as recited in claim 18, wherein said update triggering schedule includes triggering updates in a sequence in a partitioned computational nodes C_0, C_1, C_2, ..., C_N of at least two subsets and in a sequence in a partitioned computational nodes D_0, D_1, D_2, ..., D_N of at least two subsets.

21. (Previously Presented) The apparatus as recited in claim 18 wherein said sequence of turbo encoded data symbols includes "N" number of symbols, wherein each symbol in said sequence is identified by either a subscript "i" or "k" corresponding to the subscripts used for said state nodes and said computational nodes.

22. (Previously Presented) A processor configured for decoding a sequence of turbo encoded data symbols for communication over a channel comprising:

channel nodes R_x, R_y and R_z for receiving channel output;

symbol nodes X_i, Y_i and Z_k in communication with said channel nodes R_x, Ry and R_z;

state nodes S_i and S_{i-1} associated with a first constituent code in a turbo code;

state nodes σ_k and σ_{k-1} associated with a second constituent code in said turbo code;

a computational node C_i in communication with said symbol nodes X_i and Y_i;

a computational node D_k in communication with said symbol nodes X_i and Z_k, wherein said computational node C_i is in communication with said state nodes S_i and S_{i-1} and said computational node D_k is in communication with said state nodes σ_k and σ_{k-1};

a computational node C_{i+1} in communication with said state node S_i;

a computational node C_{i-1} in communication with said state node S_{i-1};

a computational node D_{K+1} in communication with said state node σ_k; and

a computational node D_{k-1} in communication with said state node σ_{k-1};

wherein computational nodes C and D at different time instances are configured for updates in accordance with an update triggering schedule, said update triggering schedule including concurrent triggering of each node of a first plurality of said computational nodes C, and concurrent triggering of each node of a second plurality of computational nodes D.

23. (Original) The processor as recited in claim 22 wherein said update triggering schedule includes triggering updates of said computational nodes C and D in a sequence of C_0, C_1, C_2, ..., C_N, C_{N-1}, C_{N-2}, C_{N-3}, ... C_2, C_1, C_0, D_0, D_1, D_2, ..., D_N, D_{N-1}, D_{N-2}, D_{N-3}, ... D_2, D_1, D_0.

24. (Original) The processor as recited in claim 22 wherein said sequence of data includes "N" number of symbols, wherein each symbol in said sequence is identified by either a subscript "i" or "k" corresponding to the subscripts used for said state nodes and said computational nodes.

25. (Canceled)

EVIDENCE APPENDIX

No evidence has been submitted pursuant to 37 C.F.R. §§ 1.130, 1.131, or 1.132. No evidence has been entered in the record by the Examiner and relied upon by Appellants in this Appeal.

RELATED PROCEEDINGS APPENDIX

Appellants, Assignee, and the undersigned atorney do not know of any other appeal, interference, or judicial proceeding that is related to, directly affects, is directly affected by, or has a bearing on the decision of the Board of Patent Appeals and Interferences in this Appeal.